

XVI INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS  
ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, ШУМЕН 2025

**Задача A22. Цикли**

 1 s  256 MB

Ян е смело момче, което винаги минава по най-краткия път от училище до дома на баба си. За съжаление, този път води директно през обитавана от духове гора.

Един ден, докато се разхождаше из гората, Ян се натъкна на ужасяващото същество Tung Tung Sahur. Съществото го предизвика на игра: ако Ян победи, ще му позволи да продължи пътя си – и дори може да получи награда. Но ако загуби, Tung Tung Sahur ще го изяде.

Играта започва с Tung Tung Sahur, който избира скрита пермутация  $P$  с  $N$  елементи. Сега е ред на Ян да играе. Той може многократно да избере две различни позиции в пермутацията и да помоли Tung Tung Sahur да размени елементите, които са на тези позиции. Tung Tung Sahur ще извърши размяната и след това ще обяви броя на **циклите**<sup>1</sup> в актуализираната пермутация. Тези размени са постоянни – Tung Tung Sahur не ги връща (но Ян може свободно да повтори размяна, за да я върне, ако е необходимо).

Във всеки момент Ян може да обяви, че пермутацията е сортирана, като извика “Sorted.” Ако пермутацията наистина е сортирана във възходящ ред, Tung Tung Sahur го пуска. Освен това ще го възнагради с няколко златни монети, на които общата стойност зависи от това колко размени са били необходими. Следователно Ян трябва да се опита да сортира пермутацията, като използва възможно най-малко размени.

Вашата задача е да помогнете на Ян да сортира скритата пермутация, като използва само дадената информация. Опитайте да го направите, като минимизирате броя на необходимите размени.

Подробности по имплементацията

Вие трябва да имплементирате функцията:

```
void sortPermutation(int N);
```

Тя ще се извиква веднъж за тестов случай с  $N$  – броят на елементите в пермутацията, която Ян трябва да сортира. От тази функция (и други функции, които пишете), можете да извикате функцията `performSwap`, която ще размени 2 елемента от пермутацията и на свой ред ще върне общия брой цикли в получената пермутация. Трябва да гарантирате, че след като вашата функция приключи, скритата пермутация е сортирана във възходящ ред  $P_i = i$  за всяко  $0 \leq i < N$ , тъй като Ян незабавно ще извика “Sorted” след връщане от тази функция.

```
int performSwap(int x, int y);
```

Извикването на `performSwap` води до размяна на елемента  $P_x$  с  $P_y$ . Имайте предвид, че ако извикате тази функция с един и същ елемент, т.е.  $x = y$ , ще получите грешен отговор. Тази функция получава  $x$  и  $y$  като индекси, започващи от 0 и връща

<sup>1</sup>Всяка пермутация на целите числа  $\{0, 1, \dots, N - 1\}$  може да бъде разбита на колекция от несвързани цикли. За да намерите един такъв цикъл, започнете от произволен индекс  $i$  и следвайте съответствието  $i \mapsto P_i \mapsto P_{P_i} \mapsto \dots$ , докато се върнете към  $i$  – те образуват един цикъл. Например пермутацията  $[1, 2, 0, 5, 4, 3]$  представлява съответствието  $0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 0, 3 \mapsto 5, 4 \mapsto 4$  и  $5 \mapsto 3$ , което води до несвързани цикли  $(0\ 1\ 2)$ ,  $(3\ 5)$  и  $(4)$ , така че има 3 цикъла.

## XVI INTERNATIONAL ADVANCED TOURNAMENT IN INFORMATICS ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА, ШУМЕН 2025

броя на циклите в **променената** пермутация.

Вашият код ще бъде компилиран заедно с грейдер, така че не трябва да съдържа `main` функция, нито трябва да чете от `stdin` или да пише на `stdout`. Трябва също да включите хедъра `cycles.h`. Пермутацията е фиксирана преди извикването на `sortPermutation` и така грейдърът не е адаптивен.

### Локално тестване

За да тествате вашата програма локално, са Ви предоставени локален грейдър и хедърен файл. Локалният грейдър чете първо  $N$  и след това  $N$  различни числа от 0 до  $N - 1$ . След това извиква вашата функция `sortPermutation` и се очаква тя да сортира правилно пермутацията.

### Ограничения

- $N = 1000$

### Подзадачи

Подзадача	Точки	Ограничения	Допълнителни ограничения
1	10	$N = 1000$	За четни $i$ : $(P_i, P_{i+1}) = (i, i + 1)$ or $(i + 1, i)$ .
2	20	$N = 1000$	Началната пермутация може да бъде сортирана точно чрез една размяна.
3	70	$N = 1000$	-

Получавате част от точките за дадена подзадача, само ако преминете правилно всички тестове в нея. Резултатът ви ще бъде пресметнат, ползвайки  $Q$  - максималния брой извиквания на функцията `performSwap` сред всички тестове:

$10^7 < Q$	0%
$10^6 < Q \leq 10^7$	10%
$9 \cdot 10^4 < Q \leq 10^6$	20%
$3 \cdot 10^4 < Q \leq 9 \cdot 10^4$	60%
$Q \leq 3 \cdot 10^4$	100%

### Примерен тест

Вход	Взаимодействие
3 2 0 1	<code>sortPermutation(3)</code> <code>performSwap(0, 1): return 2</code> <code>performSwap(0, 1): return 1</code> <code>performSwap(0, 1): return 2</code> <code>performSwap(1, 2): return 3</code> <code>sortPermutation(3): returns</code>